A Temporal Bayesian Network Reliability Framework

Hichem Boudali
Dept. of Elec. and Comp. Engineering
University of Virginia
Charlottesville, VA 22903, USA
hb4m@virginia.edu

Joanne B. Dugan
Dept. of Elec. and Comp. Engineering
University of Virginia
Charlottesville, VA 22903, USA
jbd@virginia.edu

Abstract

We explore the usage of Bayesian networks (BNs) for reliability modeling and analysis of dynamic systems. Dynamic system components exhibit complex behaviors and interactions, making combinatorial models inappropriate to solve them. Markov chains, and their extensions, have been widely used to model such systems. However, the infamous state space explosion problem greatly limits their application. We propose a novel reliability modeling and analysis framework based on temporal BNs. Three main aspects are being investigated during the development of our BN reliability framework: Modeling power capabilities (*i.e.* behavior and interaction between components), handling general component failure distributions and efficient methods to solve large dynamic systems and dealing with the state space explosion problem.

1 Introduction

The motivation of this work is twofold: (1) Alleviate the state space explosion problem encountered during Markov Chains (MCs) generation, (2) expand the modeling power capabilities of reliability tools, such as the Galileo/ASSAP tool (Sullivan et al. 1999), and at the same time keeping a high level description of those capabilities. MCs are the state of the art in analyzing dynamic systems. Formalisms such as Stochastic Petri Nets (SPNs) and Dynamic Fault Trees (DFTs) are a high level description of the system for which a MC is generated to compute performance measures. DFTs (Dugan et al. 1992) extend standard fault trees to model dynamic systems. They define a set of constructs (generically called gates in Galileo), including the warm spare gate (WSP) and the functional dependency gate (FDEP), to capture components sequential and functional dependencies.

Bayesian networks are based on a well-defined theory of probabilistic reasoning, and provide a strong framework for handling probabilistic events. They have proven to be a powerful formalism to express complex dependencies between random variables. Recently, their popularity started to grow among system reliability analysts. Earlier work (Bobbio et al. 2001; Almond 1992) has examined the parallels between BNs and FTs; and showed the obvious superiority of BNs over FTs in terms of modeling and analysis capabilities. A Bayesian network is a Directed Acyclic Graph (DAG) comprised of nodes and arcs. Nodes represent Random Variables (RV) and directed arcs between pairs of nodes represent dependencies between the RVs. Nodes without parent nodes are called root nodes and possess a prior probability distribution table. All other nodes are intermediate nodes and each one possesses a Conditional Probability Table (CPT). Nodes without children nodes are also called leaf nodes. A Bayesian network uniquely defines a joint probability distribution over all the RVs present in the graph (Charniak 1991; Pearl 1988). Virtually any probabilistic query (*e.g.* probability of RVs X and Y to be in state x and y respectively) can be answered knowing the joint probability distribution. This distinctive characteristic of BN allows us to perform, in addition to reliability computation, various other types of analyses, such as sensitivity and diagnosis.

We have illustrated our methodology using two examples, and preliminary results show that a temporal BN-based reliability framework is a promising methodology to modeling and analysis of complex dynamic systems. The first example is the Hypothetical Cardiac Assist System (Vemuri et al. 1999) and shows how its corresponding DFT is automatically converted into an equivalent BN and different analyses are conducted.

The second example illustrates the state space explosion problem encountered in MC based solutions. The DFT is rather simple, however the presence of two cascaded Priority-AND gates, *i.e.* PAND (Fussell et al. 1976), and the relatively high number of basic events makes the corresponding MC considerably large and infeasible to solve in timely fashion.

Temporal dependencies (*e.g.* a cold spare fails only after its primary has failed) are inherent in dynamic systems, and therefore finding an explicit and suitable representation of time in a BN is primordial. This is the topic of the next section.

2 A discrete-time Bayesian network reliability modeling/analysis formalism

We have developed a discrete-time BN formalism for modeling dynamic systems. Each node (or random variable, *i.e.* RV) in the BN represents a system component. A system component can be either a basic component or a subsystem describing the interaction between a collection of system components. Typically in a DFT, a basic component would be a basic event (BE) and a subsystem would be represented as a gate (*i.e.* describing some interaction between its inputs and its output). The states of a RV are the failures of the corresponding system component at consecutive points in time. A point in time is represented as an infinitesimal small time interval Δ along the time line.

Initially, we have translated a DFT into an equivalent discrete-time BN. Each basic event of the DFT is represented as a root node. In some circumstances, when dependencies exist between basic events (*e.g.* a functional dependency of a cold spare upon its primary), a basic event may no longer be a root node and thus possesses a CPT. All gates (including the top system failure gate) in the DFT are intermediate nodes in the BN and possess a CPT associated with each one of them. Conversion of the DFT into a BN is just a first step in defining our BN reliability modeling formalism. The next step is to extend the set of the DFT gates to include other constructs. Constructs are essentially a way to define types of components' behaviors and dependencies between components. Some of these extensions will include mutually exclusive events, deterministic time delays, inhibition, etc. The process of converting a DFT into a BN can be automated. Indeed, the BN graph closely maps the DFT graph and the CPTs and prior probability tables are populated according to the failure probability distributions of the basic (root) events and the types of gates tying the various system components together. Ultimately one would directly specify the dynamic system as a BN model, bypassing the DFT to BN conversion.

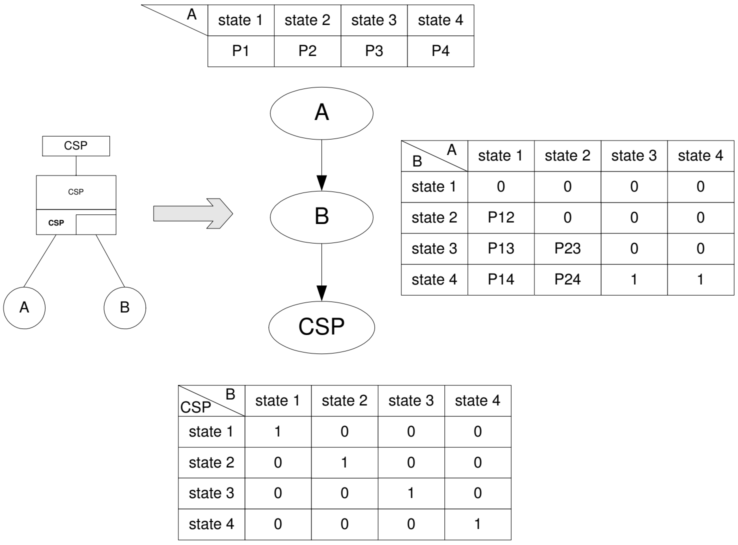
2.1 Time representation

A node, or RV, in the BN represents the state of a basic component or the state of a subsystem. The state of the gate's output reflects the state of the subsystem; or in other words, a subsystem is represented by a gate[1]. The RV X is the *event* of basic component X or gate output X to fail. The RV X is in state x (*i.e.* $X = x$) means that the basic component or the gate fails in time interval x.

We divide the time line, from time $t = 0$ till time $t = T$ (T is mission time), into n intervals. n is the time granularity. Each node variable has a finite number of states. The n first states represent the failure of component X in one of the n time intervals (*i.e.* during the mission time); and the last state, *i.e.* state $n + 1$, represents the survival of X for the duration of the mission time. The following must hold for any RV X: $\sum_{x=1}^{n} P(X = x \mid pa(X)) = 1 - P(X = n + 1 \mid pa(X))$; where $pa(X)$ is the set of all parent nodes of X whose states are known. If T is mission time and n is the time granularity. Then the mission time interval is divided into n infinitesimally small time intervals of length $\Delta = T/n$, and $P(X = x) = P(X$ fails in time interval $](x-1)\Delta, x\Delta])$. For a node X representing a basic (root) event, $P(X$ fails in time interval $](x-1)\Delta, x\Delta])$ is easily derived from the BE failure distribution $F(t)$.

As an example, let's look at the equivalent BN of a cold spare gate (figure 1). For the sake of clarity, we chose $n = 3$. *State 1* is failing in $]0, \Delta]$, *state 2* is failing in $]\Delta, 2\Delta]$, *state 3* is failing in $]2\Delta, 3\Delta = T]$, and *state 4* is not failing during the mission time. A is a primary unit and B is a cold spare unit.

[1] Gate (alternatively called construct) is used here as a generic word for describing any subsystem reflecting a certain interaction between a set of system components.



A	state 1	state 2	state 3	state 4
	P1	P2	P3	P4

B \ A	state 1	state 2	state 3	state 4
state 1	0	0	0	0
state 2	P12	0	0	0
state 3	P13	P23	0	0
state 4	P14	P24	1	1

CSP \ B	state 1	state 2	state 3	state 4
state 1	1	0	0	0
state 2	0	1	0	0
state 3	0	0	1	0
state 4	0	0	0	1

Figure 1: The CSP and its equivalent BN.

In figure 1, $P\alpha$ of the primary basic event A is the probability of A failing in state α. In principle, the $P\alpha$'s can take on any value as far as they sum up to 1. However, in most cases, a particular failure distribution $F(t)$ is given. In this case $P\alpha = P(A = \alpha) = F(\alpha\Delta) - F((\alpha - 1)\Delta)$, for $1 \leq \alpha \leq n$, and $Pn{+}1$ $= 1 - \sum_{\alpha=1}^{n} P(A = \alpha)$. This allows us to populate the prior probability distribution table of A.

The arc from node A to node B shows the dependency of B upon the state of A. B's CPT quantifies this dependency. $P\alpha\beta$ is the probability of B to be in state β given that A is in state α, *i.e.* $P(B = \beta \mid A = \alpha)$. The zero entries in the table state that the spare unit B can not fail before, or at the same time as, the primary A, *i.e.* $P\alpha\beta = 0$ for $\beta \leq \alpha$. Since B is a standby unit, we have, for all $\alpha < \beta \leq n$, $P\alpha\beta = P(B = \beta \mid A = \alpha) = G(\beta\Delta - \alpha\Delta) - G((\beta - 1)\Delta - \alpha\Delta)$; where $G(t)$ is the failure distribution of B if B were taken in isolation. Furthermore, for any given α, $P\alpha n{+}1 = 1 - \sum_{\beta=1}^{n} P(B = \beta \mid A = \alpha)$.

The CPT of the CSP output gate simply states that the CSP output fails with a probability 1 as soon as the spare unit B fails. Once the structure of a BN is known and all the probability tables are filled, it is straight forward to compute the resulting probabilities of being in each state for any of the node variables. In particular, we are interested in the leaf node variable, which represents the overall system state (*i.e.* the top gate in a fault tree), and its probability of being in the state $n + 1$.

The CPTs can be viewed as multi-dimensional tables. As a general rule the dimension d of the table is equal to the number of incoming arcs (*i.e.* number of parents) plus one, d = 1+{number of parents}. In fact, there isn't a unique BN for a given DFT; however, for the sake of bounding computation time during inference, the dimension d should be kept as small as possible for all CPTs. At this stage, we have defined the equivalent BNs for all the gates present in the Galileo DFT tool.

References

Almond, R. G. (1992). An extended example for testing graphical-belief. Research Report 6, Statistical Science Inc.

Bobbio, A., L. Portinale, M. Minichino, and E. Ciancamerla (2001, March). Improving the analysis of dependable systems by mapping fault trees into bayesian networks. *Reliability Engineering and System Safety 71*(3), 249–260.

Charniak, E. (1991). Bayesian networks without tears. *AI Magazine 12*(4), 50–63.

Dugan, J. B., S. J. Bavuso, and M. A. Boyd (1992, September). Dynamic fault-tree models for fault-tolerant computer systems. *IEEE Transaction on Reliability 41*(3), 363–377.

Fussell, J. B., E. F. Aber, and R. G. Rahl (1976). On the quantitative analysis of priority-and failure logic. *IEEE Transaction on Reliability R-25*(5), 324–326.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann.

Sullivan, K. J., J. B. Dugan, and D. Coppit (1999, June). The galileo fault tree analysis tool. In *29th Annual International Symposium on Fault-Tolerant Computing*, pp. 232 –235.

Vemuri, K. K., J. B. Dugan, and K. J. Sullivan (1999). Automatic synthesis of fault trees for computer-based systems. *IEEE Transaction on Reliability 48*(4), 394–402.